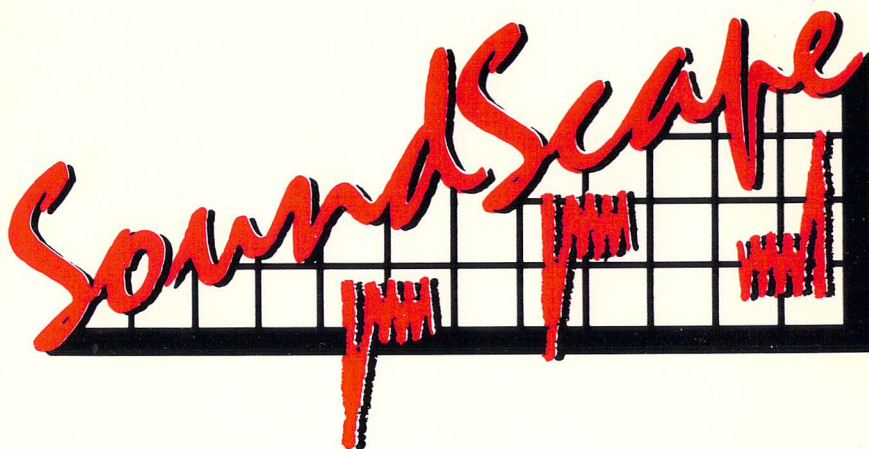




Utilities 1

mimeticsTM
corporation

...the Professional Software Source!!

COPYRIGHT

This manual Copyright ©1987, Mimetics Corporation, All Rights Reserved. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior written consent, in writing, from Mimetics Corporation.

All software described in this manual Copyright © 1987, Mimetics Corporation, All Rights Reserved. The distribution and sale of these products are intended for the use of the original purchaser only. Lawful users of these programs are hereby licensed only to read the programs, from their media into memory of a computer, solely for the purpose of executing the programs. Duplicating, copying, selling, or otherwise distributing these products is a violation of the law.

DISCLAIMER

MIMETICS CORPORATION MAKES NO WARRANTIES, EITHER EXPRESSED OR IMPLIED, WITH RESPECT TO THE PROGRAMS DESCRIBED HEREIN, THEIR QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. THESE PROGRAMS ARE SOLD "AS IS". THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT THE CREATORS OR THE PROGRAMS, MIMETICS CORPORATION, THEIR DISTRIBUTORS, OR THEIR RETAILERS) ASSUMES THE ENTIRE COST OF INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE PROGRAMS EVEN IF IT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME LAWS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITIES FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY.

SoundScape, MouseBender, Mapper/Splitter, Fuel Gauge, ClockScaler, SystemX, FrameCounter, Install Modules Mimetics are trademarks of Mimetics Corporation

Amiga and WorkBench are trademarks of Commodore-Amiga, Inc.

Commodore is a registered trademark of Commodore Electronics Limited.

TABLE OF CONTENTS

Table Of Contents	2
Features	3
Getting Started	4
Install Modules	6
MouseBender	9
Mapper/Splitter	13
Fuel Gauge	21
ClockScaler	24
SystemX	26
FrameCounter	29
SmoothClocker	32

FEATURES

This disk includes additional modules designed to enhance the use of SoundScape from Mimetics Corporation. Please note that SoundScape is required in order to use these modules - the modules alone serve no function. Also be aware that more than 512k of memory may be required to run certain modules or multiple modules, depending upon your application.

The following modules are included on the Utilities 1 Disk:

MouseBender converts mouse movements from the X- and Y-axis to user-specified MIDI information. Uses include pitch-bending Sound Sampler voices.

Mapper/Splitter transforms or "maps" any specified MIDI information to any other type of MIDI information. Also used to split the output of a MIDI device such as a keyboard into zones which transmit on different MIDI channels on different MIDI devices.

SystemX saves and retrieves MIDI system exclusive data such as synthesizer patches, samples and drum machine patterns.

Fuel Gauge displays memory usage of internal and external RAM.

ClockScaler provides user-definable clock ratios for interfacing with external MIDI clocks while using SoundScape at various metronome resolutions.

FrameCounter creates a "hit list" for a SoundScape Tape Deck file in seconds and frames for use in film composition.

SmoothClocker replaces the original Clock module with improved resolution and accuracy.

In addition, the Utilities also comes with the Install Modules program.

Install Modules allows the user to install selected modules in the start-up sequence so that SoundScape always boots with those modules in the Patch Bay. The start-up sequence can also be configured to change the default drives for data and modules to those specified by the user.

Getting Started

THE MODULAR CONCEPT

As you probably know, SoundScape is not simply a program but a series of independent programs running under the Amiga's multi-tasking operating system. The programs act much like discrete components in a recording studio which are interconnected to fulfill a desired task. These modules or software components communicate MIDI information and are available in SoundScape's Patch Panel. The mouse is used to connect "sources" from the left column of the Patch Panel to "destinations" in the right column.

This disk contains additional modules which can also be placed in the Patch Panel and connected in the same way. You might think of them as additional tools in your musical tool kit or as new components in your studio. In many cases these modules eliminate the need for "little black boxes" which are often found in MIDI set-ups.

As with SoundScape itself, the possibilities are endless. This manual covers the basic operation of the included modules and provides several examples for the possible use of each. Even if it were possible to cover all the possible permutations, it would span several volumes. Every individual has his or her own unique application and we challenge you to build wonderful things with these new additions to your "MIDI Erector Set".

FIRST STEPS

Follow these steps in order to access the modules on the Utilities1 disk:

- 1) Boot using Kickstart and your SoundScape Workbench.
- 2) Insert the SoundScape master disk in df1: and double-click on the SoundScape disk icon.
- 3) Double-click on the SoundScape program icon to enter SoundScape. You should see SoundScape's Patch Panel.
- 4) Remove the SoundScape master disk and place the Utilities1 disk in df1:.
- 5) Double-click on the SoundScape Utilities1 disk icon. The SoundScape Utilities1 window should now reveal the icons for the modules and the installation program.

(Note: If you are using a single drive system, simply load SoundScape as usual, then place the Utility disk in df0: to access the modules.)

USING A MODULE

To temporarily install a module in the Patch Panel, simply double-click on the module's icon in the SoundScape Utilities1 window. The module will then appear in the Patch Panel. Some modules can be added to the Patch Panel more than once, depending on the user's needs.

In most cases, the new module can now be patched to other modules using the traditional method of single-clicking in succession on the modules to be connected. Where applicable, windows for the modules can also be opened by double-clicking on their Patch Panel icons. All requestors and Load and Save display windows operate the same way as the ones in SoundScape you are already familiar with.

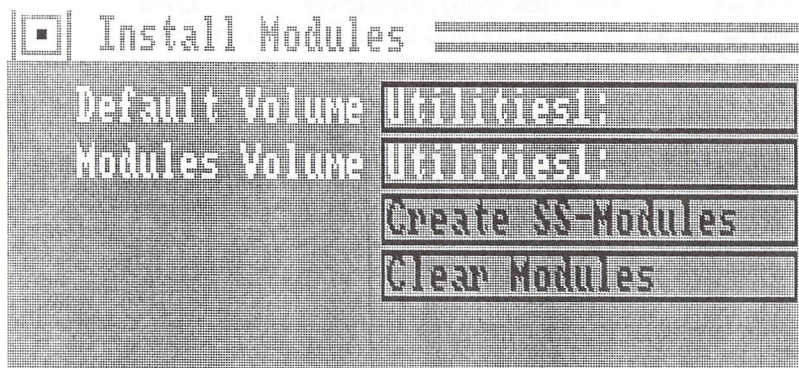
Utility modules can also be permanently installed in the Patch Panel. See the section on Install Modules.

USING THE EXAMPLES

In order to use the examples, first make sure that the module in question is loaded into the Patch Panel. This is not necessary after modules are permanently installed on your Workbench. The examples themselves are SoundScape Environments which are loaded in the normal way. Refer to the example under Install Modules (next section) and change your Default: drive to Utilities1: as specified to insure accurate loading of the examples.

After you finish trying one example environment, remove the Patch Panel connections before going to the next example as patches are not removed automatically.

Install Modules



FUNCTION AND USES

Install Modules helps configure your SoundScape Workbench so that SoundScape boots with the modules of your choice in the Patch Panel. Install Modules also allows the user to configure which drives SoundScape treats as defaults for data and modules.

PATCHING

Install Modules does not appear in the Patch Panel. Instead a window opens temporarily with the appropriate commands.

OPERATION

Double-click on the Install Modules icon to open the corresponding window. Install Modules basically creates a file in the s directory called "SS-modules" which SoundScape refers to when booting to determine default system attributes. Install Modules only works properly with SoundScape 1.4 and subsequent revisions.

Default Volume

The top box determines where SoundScape defaults for data storage and retrieval. Before any changes, the program defaults to the SoundScape: disk for storage and retrieval of tracks, samples, environments, etc.. Changing to another disk name or drive/path

has the same effect as the DOS Assign command - data will be redirected to the new source/destination. If you specify df0: or df1: as the default volume, SoundScape will select the disk in the analogous drive as it boots. It is therefore recommended that volume names be used instead or that the disk you wish should be in the drive. You may find it advantageous to create a custom SoundScape Workbench for each project, using this volume as the default to hold your data.

Click on the Default Volume requestor and enter the desired default device. After any other changes have been made, click on Create SS Modules to install this change (see below). Note that the system will have to be rebooted in order for the installation to be effective.

Modules Volume

The second box determines where SoundScape looks for modules that are included in the start-up sequence. The program looks for additional modules on the volume specified in the requestor. As with Default Volume, changing the value to another disk name redirects the search for modules to the desired drive/path or device.

Click on the window and enter the desired default device. Click on Create SS Modules after any other adjustments have been made to install this change (see below).

Create SS Modules

Single click on this box to install any modules which are currently in the Patch Panel into the "s/SS-modules" boot file on your SoundScape Workbench. This command also configures the file to assign the Default Volume and Modules Volume to any changes that you have made.

Caution: Be sure that if you install modules into the "s/SS-module" file that there is enough space on "Modules Volume" to place the module itself and make certain to move a copy of the module there (assuming you move them from the Modules: disk). Otherwise, SoundScape will encounter a fatal error on re-booting. Under no circumstances should you alter the SoundScape disk itself. The SoundScape Workbench or a sub-directory on a hard drive are therefore prime candidates for relocation of permanently installed modules.

The real power of Install Modules is in creating self-configuring set-ups for given applications. Custom SoundScape Workbench disks can be made for each project so that each SoundScape application has a turnkey disk which includes load custom modules and environments complete with default setting for data storage.

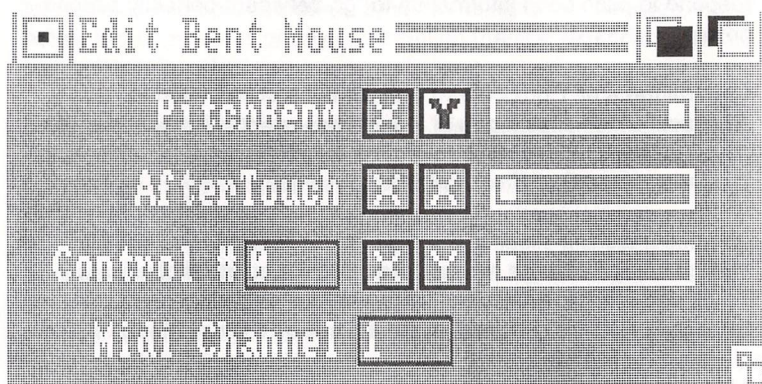
Clear Modules

Single click on this box to remove all modules which are not native to SoundScape from the Patch Panel. Please note that this does not change the "s/SS-modules" file itself. The file can also be edited as a text file to make any deletions or to automate additional aspects of SoundScape when the system is booted.

EXAMPLES

Our first example will help you use the rest of the examples in this manual! All the data for the examples (environments, samples, tracks, etc.) is stored on the Utility disk. What we want to do is change the default data device to this disk. Open the Install Modules window and change Default: from "SoundScape:" to "Utilities1:" (no space, don't include the quotes and do include the colon). Then click on Create SS Modules to make the change. You will need to reboot because SoundScape only knows about the change as a result of the booting process. After going through the examples, you'll probably want to change the Default: to something else for your projects.

MouseBender



FUNCTION AND USES

MouseBender translates mouse movements along the X- or Y- axis or both into MIDI information such as pitch bend, aftertouch and MIDI controller data.

Typical uses of MouseBender include bending the pitch of a Sound Sampler instrument, riding the volume of a track or changing any parameter on a MIDI device which can be assigned a MIDI Controller.

PATCHING

MouseBender comes up on the Patch Panel as a MIDI source and can be patched to any of the destinations in the right column of the Patch Panel, depending on your application.

OPERATION

Once you've installed MouseBender in the Patch Panel, double-click on the new icon to open the MouseBender window. One or more of the following features can be engaged simultaneously. MouseBender parameters are stored and recalled as part of SoundScape Environments.

PitchBend

Many synthesizers and MIDI devices including the Sound Sampler respond to pitch bend information to "go between" perfectly pitched notes.

Click on the X or Y box or both with the left mouse button to determine which direction(s) of mouse movement will result in a pitch change. The X-axis represents horizontal (side-to-side) motion and Y represents vertical (front-to-back) movement.

Drag the matching slider to the appropriate position to select the amount of pitch change as compared to the amount of physical movement in the mouse. (Note: There will be no change in pitch if the slider is all the way to the left or if neither X or Y is selected.)

Aftertouch

Aftertouch refers to the amount of pressure exerted on the keyboard after keys have been depressed. While aftertouch is often routed to effect volume, many MIDI devices allow aftertouch to effect other parameters. Note that aftertouch is not implemented on all MIDI devices and is not implemented on Sound Sampler.

Clicking on the X and/or Y boxes with the left mouse button selects which axis effects a change in aftertouch. The X-axis signifies horizontal movement and the Y-axis determines vertical movement.

Dragging the X- and Y- sliders selects the amount of aftertouch change relative to the amount of physical movement of the mouse. (Note: If the slider is all the way to the left or if neither X or Y is selected, there will be no change in aftertouch.)

Controller #

Some MIDI devices receive MIDI Controller information such as data from wheels, levers, sliders, pedals and other performance devices - in this case it's a mouse! These Controllers then are routed to various functions on the destination device.

Click on the first box and enter the MIDI Controller # (0-127) that you wish the mouse to effect. As an example Controller #64 is often used to communicate sustain pedal information. Consult the documentation on the external gear you are using to establish

which Controllers are implemented and their numbers.

Click on the X or Y box or both with the left mouse button to select which mouse movements will result in a change in Controller status. As with pitch bend and aftertouch, the X-axis is analogous to horizontal movement and the Y-axis is analogous to vertical motion.

The matching sliders determine the amount of change on the selected MIDI Controller as compared to the amount of physical movement in the mouse. (Note: There will be no Controller change if the slider is all the way to the left or if neither X or Y is selected.)

Midi Channel

The output from the MouseBender module can be transmitted on any of the 16 MIDI channels.

Click on the Midi Channel box and enter the desired output channel (1-16).

Once the parameters in the MouseBender window have been set to your needs, click on the window's Close Box. To actually engage MouseBender, click once on the small Mouse window in the lower left corner of the screen to activate MouseBender. Move the mouse from side to side to effect parameters selected for the X-axis and move it forward and backward to effect the parameters selected for the Y-axis. To disengage MouseBender, click on any other SoundScape window.

EXAMPLES

Example 1 - Bending The Pitch Of A Sound Sampler Instrument

For the purposes of this example, the Console Keyboard is used to play a Sound Sampler instrument on MIDI Channel 1. MouseBender is also patched to Sound Sampler with the appropriate setting to enable pitch bending of the internal instrument. Vertical movement of the mouse changes the pitch. A note must be played from the Console Keyboard in order to hear the pitch change.

Example 2 - Adding Pitch Bend To A Tape Deck Track

The first track, TrackToBend, has a pre-recorded performance which is routed to the Mixer on Channel 1. The second track, Pitch Bend, is

recording pitch changes from MouseBender and is also passing them on to the Mixer on Channel 1. The third track, Composite, is recording the output of the Mixer which contains the outputs of both TrackToBend and Pitch Bend and passed Thru to the Sound Sampler. Note that Pitch Wheel is enabled on the Status In and Status Out boxes of the Pitch Bend and Composite tracks. Vertical movement of the mouse changes the pitch.

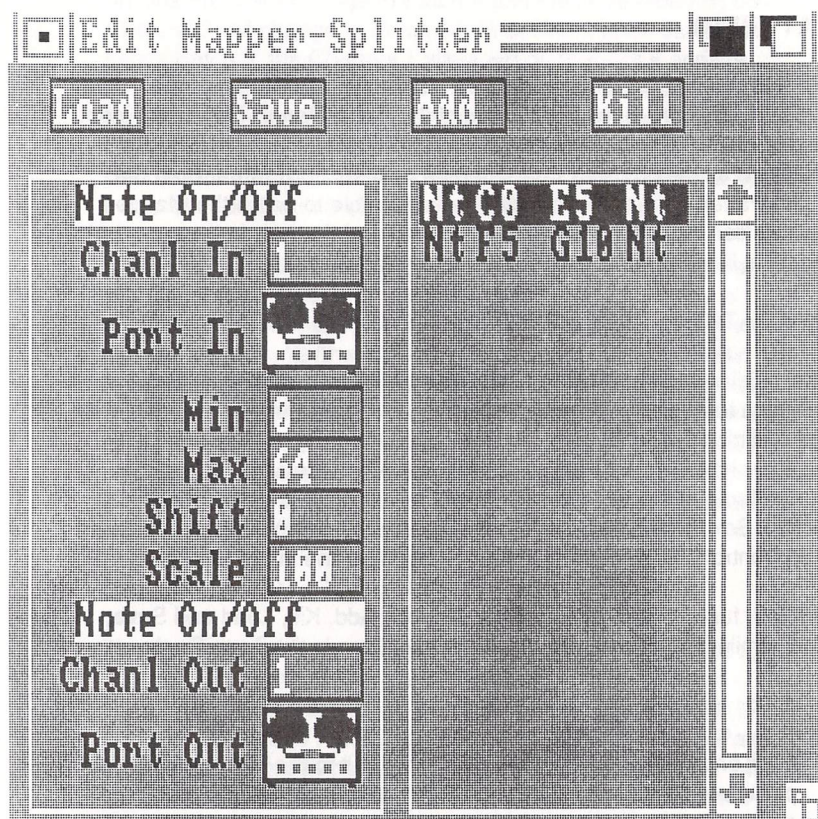
In the above situation, a separate track was recorded for pitch changes and the two tracks were then combined into a third track.

Example 2A shows a short-cut where MouseBender is being routed directly to the Mixer.

Example 3 - Changing The Tempo With MouseBender

In this example "TrackToClock" has a pre-recorded performance. A second track, "Clock Change" is recording Controller # 1 from horizontal movement of MouseBender and sends it to the Clock which is driving the Tape Deck as usual. The Clock sees Controller # 1 as a tempo change. Note that Control Change is enabled in the Status In and Status Out boxes of the "Clock Change" track.

Mapper/Splitter



FUNCTION AND USES

Mapper/Splitter is a powerful tool which allows you to map or transform any type of MIDI data into any other type of MIDI data. This module also provides for multiple maps simultaneously.

Mapper/Splitter can be used to split an input device into ranges or zones, each outputting on a separate MIDI channel to other devices or modules. Mapping also comes in handy in making sure that the pitch bend range on two different instruments match up. Another use involves

translating one type of Controller like the volume pedal to meaningful data for something like a fader on a MIDI lighting console or MIDI mixing board.

PATCHING

The Mapper/Splitter module must be patched to receive from one of the available sources in the left column of the Patch Panel. This same input device should also be chosen as Port In (see next section). Output routing is achieved using Port Out (see next section) rather than the Patch Panel. As a matter of fact, it is impossible to patch the Mapper/Splitter icon in the left Patch Panel column and the icon is there so that modules like Tape Deck "see" it as an available input.

OPERATION

After adding Mapper/Splitter to your Patch Panel, double-click on the new icon to open the Mapper/Splitter window. This window operates with some similarities to the now-familiar Tape Deck. The right side of the window shows the Map List, while the left side is the Map Control Box which displays the current map. Like the relationship of the Track Control Box to the Track List in the Tape Deck, a map is displayed in the Map Control Box by clicking on the appropriate line in the Map List.

The four functions at the top of the window, Add, Kill, Load and Save, work similarly to Create, Delete, Load and Save in the Tape Deck:

Add

Click on Add to create a new map and add it to the Map List. The parameters for the new track default to those of the last map selected.

Kill

Select the map you wish to delete in the Map List using the mouse. Click on Kill to remove the currently selected map from the Map List.

Save

Click on Save to store the Map List and the parameters of all included maps to disk. You will be prompted for a file name and you may change the directory if desired. Select Quit instead to cancel. Note that using Save is necessary in order for Mapper/Splitter to be

saved with SoundScape Environments. Mapper/Splitter files are given the suffix ".split".

Load

Click on Load to retrieve stored Map Lists and their contents. This window operates in the same way as other SoundScape Load formats.

The parameters in the Map Control Box are used to set up function of the currently selected map. (Once again, the current map is the one highlighted in the Map List.)

Input Data Type

A map can get its input from one of 6 basic categories of MIDI data - Note On/Off, Poly Pressure, Control Change, Program Change, Mono Pressure and Pitch Wheel.

Click on the first inverse bar to cycle through the six types of input:

Note On/Off - Note On and Note Off commands including velocity (what notes are played, when, how long long and with what force).

Poly Pressure - Individual pressure exerted on each note after the initial sounding and before its release. (Poly pressure is only implemented on certain controllers).

Control Change - MIDI Controller data such as sustain pedal and mod wheel. (See individual manufacturer's specifications for MIDI Controller implementation.)

Program Change - Program change (patch change) information

Mono Pressure - Aftertouch or force exerted on a controller after the note has been sounded and before it's released. (Aftertouch is not implemented on all controllers).

Pitch Bend - Data from pitch bend devices such as wheels, levers and pedals.

Chanl In

A map can derive its input from any one of the 16 MIDI channels.

Click on the number beside Chanl In and change it to the desired MIDI channel (1-16).

Port In

A map can receive from any of the output ports currently available on the left side of the Patch Panel. For example if the MIDI In icon were selected, the map would derive it's input only from the MIDI In jack.

Click on the icon next to Port In to cycle through the port choices.

Min and Max

We take these two together because they are used in conjunction to establish an input range. So, if Data Type is Note On/Off and Min were set at 60 and Max at 72, the range would be restricted to the octave above middle C. When the Input Data Type and Output Data Type are set for Control Change these boxes change to reflect Controller Number In and Controller Number Out respectively.

Click on the numbers beside Min and Max and change the values to represent the desired range. While the permissible range is -999 to 9999, in practice Max should never be lower than Min and Min should never be higher than Max.

Shift

A map can shift a value higher or lower, particularly useful for things like tranposition and patch mapping. This might also be thought of as an offset.

Click on the number next to Shift and change it to the desired offset. While the permissible range is -999 to 9999, most applications fall within the range of -127 to 127. The default value of 0 represents no offset.

Scale

Scale controls the ratio of output amount compared to the input amount. The Scale value represents a percentage.

Click on the number next to Scale and change it to the desired

percentage. The default value of 100% represents a 1-to-1 ratio. The permissible range is -999 to 9999. (Note: with a value of 0, the output will not change regardless of other settings.)

The last three controls effect the output of the Mapper/Splitter module.

Output Data Type

The output can be mapped to any specified data type, regardless of the type of data coming into the Mapper/Splitter.

Click on the second inverse bar to cycle through the six types of output:

Note On/Off - Note On and Note Off commands including velocity.

Poly Pressure - Individual pressure for each note.

Control Change - MIDI Controller data.

Program Change - Program change (patch change) information

Mono Pressure - Aftertouch data

Pitch Bend - Data effecting pitch bend.

Chanl Out

A map can send its output on any one of the 16 MIDI channels.

Click on the number beside Chanl Out and change it to the desired MIDI channel (1-16).

Port In

A map can transmit to any of the input ports currently available on the right side of the Patch Panel. For example if the MIDI Out icon were selected, the map would send it's input only to the MIDI Out jack.

Click on the icon next to Port Out to cycle through the port choices.

EXAMPLES

Example 4 - Splitting A Pre-recorded Track

For this example a short boogie-woogie track called "Composite" has been recorded from MIDI In on Channel 2. "Composite" has a walking bass line in the left hand and choppy chord stabs in the right, both originally routed to "StretchPiano" in the Sound Sampler. That track is then routed out Channel 1 to the Splitter so that the components of the track can be separated as "Left Hand" and "Right Hand" tracks. The first map takes all note information coming from the Tape Deck on Channel 1 up through the E above Middle C (note 64) and passes it as note data back to the Tape Deck on Channel 1. Similarly, the second map takes all note information coming from the Tape Deck on Channel 2 above that E (note 65) and passes it as note data back to the Tape Deck on Channel 2. Both of these tracks are set to record and Thru is used to hear the results while recording.

The original piano track is now split so that "Left Hand" still plays the "StretchPiano" sample but "Right Hand" plays the "Do" sample. To hear the original piano track ("Composite"), mute "Left Hand" and "Right Hand" and redirect the output icon of "Composite" to the Sound Sampler.

Example 4A - Recording Notes and Pitch Bend On Different Tracks

It is sometimes desirable to record notes and other performance controls such as pitch bend on separate tracks. For one thing, this makes editing one particular type of data easier. In this example, the Mapper/Splitter is used to isolate these two types of MIDI data. While both maps receive their input from MIDI Channel 1, the type of input and output for the first map is restricted to Note On/Off information and the filtered data is sent to the first track in the Tape Deck on Channel 1. The second map is restricted to Pitch Bend and routed to the second Tape Deck track on Channel 2. (Note that this track has Pitch Bend enabled in Status In and Status Out.) Both tracks send on the same channel which recombines the signals on playback. The output device is left to the user's discretion.

Example 5 - Patch Mapping

This example assumes a master MIDI device is connected to

SoundScape's MIDI In jack and two MIDI slaves connected to SoundScape's MIDI Out jack. When Patch 12 is selected on the master device transmitting on Channel 1, the slave device receiving on Channel 1 will change to Patch 4 and the slave receiving on Channel 2 will change to Patch 14. In addition, when the master sends Patch 3, the Channel 1 slave will go to Patch 6 and the Channel 2 slave will go to Patch 9. (This assumes that the devices in question begin with Patch 0. Other patch configurations, such as those beginning with Patch 1 will react slightly different and require calculation of an offset. For example, Patch 1 on a Yamaha DX7 is interpreted as Patch 0 on many other MIDI devices.)

MIDI In is routed to Mapper/Splitter in the Patch Panel. In the Mapper/Splitter window for the first map, Program Change is accepted on Channel 1 from the MIDI In port. Any incoming patch change to Patch 12 (the only thing allowed by the limits of Min and Max here) is shifted down -8 to come out as Patch 4 and sent as a Patch Change on Channel 1 out the MIDI Out port. The other three maps work similarly and illustrate some of the real power of layered maps. Note that many more layers of instruments could be patch-mapped simultaneously, in addition to the mapping of additional incoming Patch Change numbers.

Example 5A - Mapping MIDI Notes To Patch Changes

This example achieves the same output results as Example 5, however Note On/Off information is being used as an input. Whenever the lowest key on a 5-octave keyboard (Note 36) is pressed, the device receiving on Channel 1 will change to Patch 4 and the device receiving on Channel 2 will change to Patch 14! The second pair changes the Channel 1 device to Patch 6 and the Channel 2 device to Patch 9 when MIDI Note 37 (C#) is depressed. This example illustrates the power of literally mapping anything to anything!

Example 6 - Split Keyboard

This example assumes a master MIDI device transmitting on Channel 1 is connected to SoundScape's MIDI In jack and two MIDI slaves are connected to SoundScape's MIDI Out jack. The object is to split the output of the master keyboard or device at Middle C, recording the two ranges on two different Tape Deck tracks with the lower notes passed on to a slave receiving on Channel 2 and the higher notes going to another slave receiving on Channel 3.

MIDI In is routed to Mapper/Splitter in the Patch Panel. The module employs two maps to attain the desired effect. Both maps receive their input as Note On/Off information on Channel 1 from the MIDI In port. The first map accepts all the notes up to the B (note 59) below Middle C and transmits that information as Note On/Off data out the Tape Deck port on Channel 2. The second map accepts all the notes above Middle C (note 60) and transmits that information as Note On/Off data out the Tape Deck port on Channel 3.

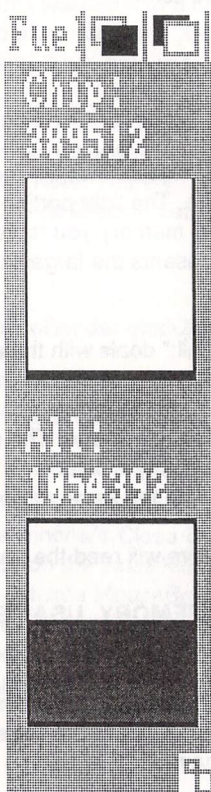
The Left Hand track of the Tape Deck is set to record from Mapper/ Splitter on Channel 2 (the lower map) and pass the performance Thru to the instrument receiving on Channel 2. The Right Hand track is set to record from Mapper/ Splitter on Channel 3 (the upper map) and pass the performance Thru to the instrument receiving on Channel 3. Note that the additional of more maps would allow for as many split points as desired.

Example 7 - Controller Mapping

This example assumes a master MIDI device is sending modulation amount from a variable footpedal (Controller 4) through SoundScape to a slave device on which we wish to add modulation (Controller 1) when the master's footpedal is depressed.

MIDI In is routed to the Mapper/ Splitter in the Patch Panel for the Controller mapping, as well as straight to the MIDI Out for the notes themselves. The map itself receives Control Changes on Channel 1 from the MIDI In port and converts Controller 4 to Controller 1. The shift can be adjusted to have the desired amount of modulation when the pedal is depressed a certain distance. The result is sent as a Control Change to the MIDI Out port on Channel 1.

Fuel Gauge



FUNCTION AND USES

The Fuel Gauge module displays available memory both for the Amiga's internal 512k and for an entire system with expanded memory.

Fuel Gauge is a useful tool when trying to cram as much information as possible into SoundScape. This is especially so when using samples extensively in the Sound Sampler.

PATCHING

Fuel Gauge is not placed in the Patch Panel. Instead its window appears in the lower-right portion of the screen.

OPERATION

After double-clicking on the Fuel Gauge icon, the Fuel Gauge window displays two indicators:

The top indicator labeled "Chip:" deals with the memory usage of the internal 512k RAM in the Amiga. The light portion of the graph indicates the total percentage of internal memory you have left since you started Fuel Gauge. The number represents the largest chunk of contiguous chip memory available.

The bottom indicator labeled "All:" deals with the usage of the all the memory online in the Amiga, including extended memory boards. The light portion of the graph indicates the total percentage of all memory you have left since you started Fuel Gauge. The number represents the largest chunk of contiguous memory available. If you are running a 512k system, the "All:" display will read the same as the "Chip:" display.

In a 512k system, both indicators will read the same.

A FEW WORDS ABOUT MEMORY USAGE AND FRAGMENTATION

A basic understanding of how the Amiga's memory is used by SoundScape will help you get the most from the system and illustrate why Fuel Gauge is handy to have.

SoundScape itself loads into external memory if available. Track data and other SoundScape information can not only take advantage of extended memory, but can also fit into available cracks in memory. On the other hand, digitized sounds in the Sound Sampler require chunks of contiguous memory in order to load or sample. Further, samples can only load into or be created in the internal chip memory at this point in time due to the Amiga's architecture. Open windows require contiguous chip memory. Finally, modules themselves will load in to expanded memory if available, while a small amount to chip memory will be used for the icons in the Patch Panel.

This should lead to several initial conclusions: 1) In a 512k Amiga, SoundScape, extra modules, open windows, track data and sound

samples all vie for the same memory. 2) In an extended memory machine, the SoundScape program, modules and track data can take advantage of extended memory while internal memory is less congested for samples and open windows. 3) When using lots of samples, lots of notes, lots of modules or any combination thereof, extended memory is highly recommended.

Taking a deeper look: 1) A sample you wish to load must be smaller in size than the number value for "Chip:". The graphic representation of the Fuel Gauge may indicate a large amount of memory available, but remember that samples require complete chunks of memory and that's what the numeric value tells you. 2) There is less memory available for samples when windows are open or enlarged. Try closing unneeded windows or making them smaller when trying to squeeze more sample data into memory. 3) Chip memory can get fragmented by repeatedly loading and deleting a variety of samples, especially in conjunction with large and/or excess open windows. The only way to fix fragmented memory is to reboot the system.

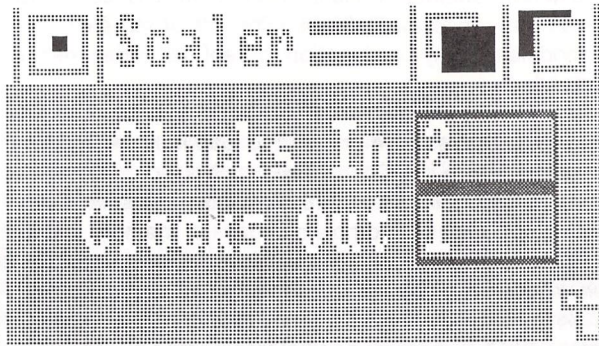
Note: SoundScape reserves the last 80k or so of memory so that you can't use it even if the indicator says it's available.

EXAMPLES

Example 8 - Low Memory

In the example, the Sound Sampler is fairly full and about 96k is left of contiguous memory. Try loading the "heythere" sample and you should be successful because it takes 13820 bytes of memory and you're still above the 80k limit. Try to load "StretchPiano" instead and you will get a low memory message and an aborted load because it's 49137 bytes long which would put you way under the 80k limit.

ClockScaler



FUNCTION AND USES

The ClockScaler module can change the ratio of the number of MIDI clock pulses at its input to the number of MIDI clock pulses being output from the module.

The MIDI clock standard is used to synchronize various MIDI devices together by dividing each quarter note into 24 pulses. This causes no problem when using SoundScape with met24 (the metronome which provides a click every 24th clock pulse), because that represents a quarter note on all MIDI devices. Many users, however, use met48 or met 96, especially to capture performance nuances like pitch bend. For example, at met96 each quarter note is sub-divided into 96 clock pulses. Unfortunately that one quarter note to SoundScape is now 4 quarter notes ($96/24=4$) to external devices using MIDI clock. ClockScaler has the ability to "scale" the ratio of clock pulses within SoundScape to the clocks in external devices to match them back up.

PATCHING

Driving An External MIDI Clock

To use the ClockScaler when driving an external device, patch the Clock source in the left column to the ClockScaler destination in the right column. Then connect the ClockScaler source in the left column to the MIDI Out destination in the right column.

Slaving To An External MIDI Clock

Usually SoundScape is driven by an external clock by using the Patch Panel to connect the MIDI In source in the left column to the Clock destination in the right column. To use the ClockScaler when slaving to an external clock, patch the MIDI In source from the left column to the ClockScaler destination in the right column. Then connect the ClockScaler source in the left column to the Clock destination in the right column.

OPERATION

After adding ClockScaler to your Patch Panel, double-click on either matching icon to open the ClockScaler Window.

Clocks In and Clocks Out

These two controls work together to form the ratio of input clock pulses to output clock pulses. Click on the numbers beside Clock In and Clock Out and enter the appropriate values.

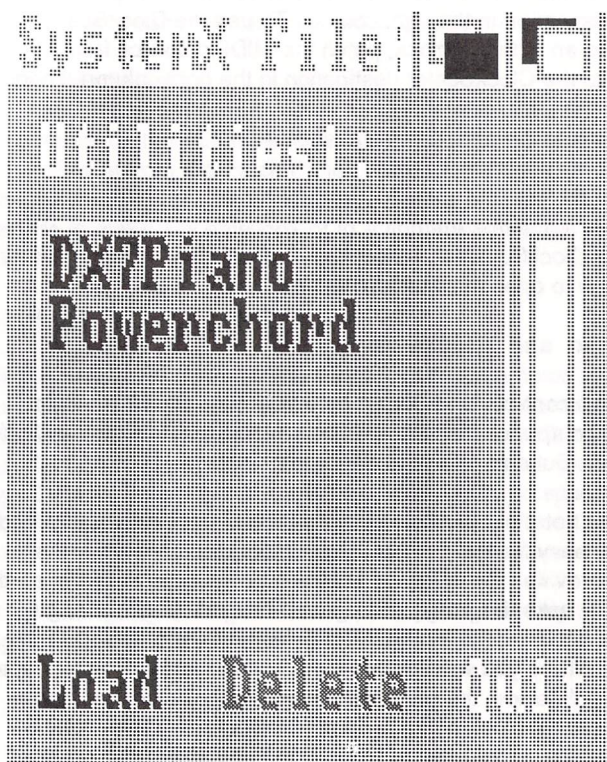
Important note: In the case of using ClockScaler as a clock divider, it is fairly easy for the program to omit clock pulses. While the module provides the ability to set the ratio as a clock multiplier, that effectively asks the program to generate extra clocks at regular intervals between the input clocks (which it cannot easily do). In an example where you want to use met96 while slaving SoundScape to an external device, the program will output 4 clock events when each incoming clock event arrives (Clocks In: 1, Clocks Out:4). Unfortunately, all events will be jammed onto the incoming clock pulse rather than having the resolution of met96. For this reason, ClockScaler is much more effective when dividing clocks to drive a slave device than it is when multiplying clocks from a master device.

EXAMPLES

Example 9 - Driving A MIDI Drum Machine While Using Met 48

Since SoundScape has twice the effective resolution of external devices when using met48, a synchronized slave device needs to see a single clock pulse for every two clocks output by the program. The settings are therefore Clocks In 2, Clocks Out 1. Clock In is routed both to the Tape Deck and the ClockScaler where the conversion is made and routed to MIDI Out for the slave device.

SystemX



FUNCTION AND USES

The SystemX module allows system exclusive data from many MIDI devices to be stored as an Amiga disk file for later retrieval.

Most MIDI synthesizers can transmit and receive the parameters which make up patches or programs via MIDI. This data is called "system exclusive" because it only has meaning to that same manufacturer's equipment (often only the same model). SystemX allows the Amiga to act as a library or storage media to replace cartridges or data cassettes. Other MIDI devices such as drum machines are also capable of communicating system exclusive data. Check the

documentation to your particular external MIDI device(s) to confirm the implementation of system exclusive data.

Note that certain devices require a "handshake" to send or receive system exclusive. SystemX does no such handshaking and can only be used with equipment which can send and receive without external prompting. Future modules will be available to communicate with devices requiring special considerations.

PATCHING

Patching within the Patch Panel is not necessary with SystemX. Simply connect the MIDI Out of the external device to SoundScape's MIDI In jack for storage and SoundScape's MIDI Out jack to the external device's MIDI In for retrieval.

OPERATION

Storage

After adding SystemX to your Patch Panel, double-click on the SystemX icon in the left column. Enter the name you wish to save the system exclusive dump as, changing the directory if necessary. Note that files are automatically tagged with the suffix ".sysex" at the CLI level. When you see the requestor "Send Me A Sys Ex File", initiate whatever procedure is required on your external MIDI device to send the system exclusive dump. To abort, press the mouse button.

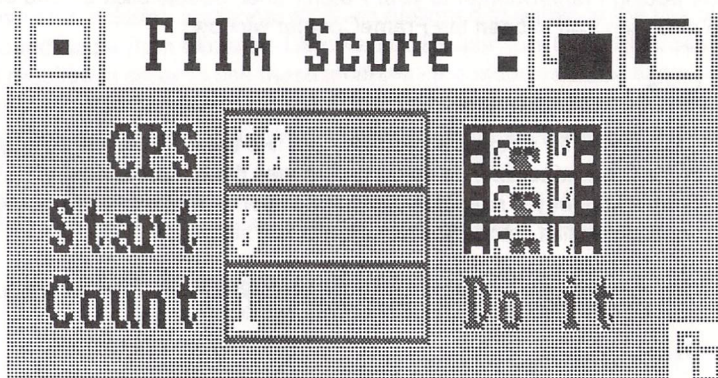
Retrieval

To retrieve system exclusive files from disk and send them back into the original device, double-click on the SystemX icon in the right column. The Load window operates in the same way as other SoundScape Load operations. Note that SoundScape uses a "disk caching" technique and sometimes newly saved data does not show up in the Load window. If this is the case, simply select the volume title and press <Return> to update the display. After selecting the file to load, you'll receive a prompt asking, "Is your Synth ready to receive?". Initiate any procedure required to tell your external device to accept a system exclusive dump and click the left mouse button on "Ready". Note that many instruments have a memory protect function that must be disengaged.

EXAMPLES

Since each MIDI device has its own protocol, examples are almost impossible! Just be sure to acquaint yourself with all the system exclusive functions of your external gear.

FrameCounter



FUNCTION AND USES

The FrameCounter module takes a given score at a given tempo and plots it out against the seconds/frames reference used in motion pictures. Essentially it maps out a "hit list" into a text file showing where each event falls, so that a composer can adjust the tempo to force certain aural events to fall on the desired frames to match a visual effect.

While FrameCounter is of value to a small percentage of SoundScape users, it is included largely to show the power of modules. The FrameCounter module more or less takes over the machine as it loads tracks into the Tape Deck, sets the Clock, steps the Tape Deck through a passage, performs the necessary calculations and generates a text file. Future film tools are being designed which fulfill more advanced needs, just as more modules are being developed in general which use the module power shown in FrameCounter. Note that FrameCounter parameters are not saved with Environments.

PATCHING

FrameCounter appears in the Patch Panel only for the purpose of opening a window. Patching is not required.

OPERATION

After adding FrameCounter to your Patch Panel, double-click on one of the matching icon to open the FrameCounter window.

CPS

CPS stands for clocks-per-second, just like the CPS tempo setting in the Clock window. This value determines the tempo at which FrameCounter will calculate the composition's playback against the unchanging film standard of 24 frames per second.

Click on the number beside CPS and change it to the desired value.

Start

Start determines how many seconds into the composition the "hit" calculations will start.

Click on the number beside Start and change it to the desired value.

Count

Count determines how many seconds of "hits" will be calculated after the Start value.

Click on the number beside Start and change it to the desired value.

Do It

Click on Do It after CPS, Start and Count have been set and you will be given a directory and prompted for a file name. Select the name of the group of tracks you wish to calculate hits for in the normal way, changing the directory as needed. A second prompt will ask the name of the text file which the calculations should be saved to. Enter the file name in the usual manner, changing the directory if desired. FrameCounter will then proceed with the analysis. Resulting files are automatically tagged with ".film" as a suffix at the CLI level.

The results of FrameCounter's analysis can be viewed (or printed) by using any text editor or entering "type [drive:path/filename]" from CLI. The horizontal axis of the display represents the 16 MIDI channels while the vertical axis represents elapsed time in seconds and frames. Experimentation with different CPS settings on subsequent passes of

FrameCounter is required to find the correct tempo setting to accommodate the majority of your hits. When things line up as desired, transfer that CPS setting to the Clock when playing back the composition in the Tape Deck. Note: FrameCounter only works properly with fixed-tempo tracks because it does each analysis at a static CPS setting. Tracks which contain a clock control track to vary the tempo will not be analyzed properly.

EXAMPLES

Example 10 - Calculating Hits For Example 4

In this example the first 20 seconds of the tracks from Example 4 have been calculated at 70 CPS and placed into a text file called "Example4At70".

Example 10A - The same 20 seconds of Example 4 have been calculated at 75 CPS and placed into a text file called "Example4At75".

SmoothClocker



FUNCTION AND USE

SmoothClocker is a direct replacement for the Clock module that is shipped with SoundScape. This newer version is much smoother and should offer much more accurate results.

PATCHING

SmoothClocker is patched exactly like the original Clock module for various applications.

OPERATION

When loaded into the Patch Panel, SmoothClocker totally replaces Clock. Simply use it as you would use the Clock.

EXAMPLES

We suggest that you install SmoothClocker permanently using Install Modules so that SoundScape always boots with this improved clocking capability.

100

The first of these is the fact that the
the second is the fact that the
the third is the fact that the

the fourth is the fact that the
the fifth is the fact that the
the sixth is the fact that the

the seventh is the fact that the
the eighth is the fact that the
the ninth is the fact that the

the tenth is the fact that the
the eleventh is the fact that the
the twelfth is the fact that the

the thirteenth is the fact that the
the fourteenth is the fact that the
the fifteenth is the fact that the

the sixteenth is the fact that the
the seventeenth is the fact that the
the eighteenth is the fact that the